2010 International Conference on Advances in Social Networks Analysis and Mining

A Framework for Improved Adolescent and Child Safety in MMOs

Lyta Penna, Andrew Clark, George Mohay
Information Security Institute
Queensland University of Technology
GPO Box 2434, Brisbane, QLD 4001 Australia
(l.penna, a.clark, g.mohay)@qut.edu.au

Abstract— **This paper presents an approach to providing better safety for adolescents playing online games. We highlight an emerging paedophile presence in online games and offer a general framework for the design of monitoring and alerting tools. Our method is to monitor and detect relationships forming with a child in online games, and alert if the relationship indicates an offline meeting with the child has been arranged or has the potential to occur. A prototype implementation with demonstrative components of the framework has been created and is introduced. The prototype demonstration and evaluation uses a teen rated online relationship-building environment for its case study, specifically the predominant Massive Multiplayer Online Game (MMO) World of Warcraft.**

Keywords- Child online safety, Internet safety, MMO, Hazardous online relationships, Meeting offline, World of Warcraft

I. INTRODUCTION

Locating and initial grooming of adolescents occurs in online games including those on PC, Xbox, Playstation and Wii. In December 2007, police arrested 19-year-old Adam Glenn Schroeder for using the PC game World of Warcraft (WoW) to lure a 12-year-old girl into having sex with him. He was convicted for criminal sexual conduct with a minor and using a computer to commit a crime, and sentenced to 10 years in prison [12]. In February 2008, 38-year-old David Faboo was arrested after allegedly abducting a mentally challenged 16-year-old girl met in WoW. He was apprehended five hours after picking up the girl near her home and found in possession of gifts for the girl as well as rope, several knives, condoms and sex toys. The back of his truck was equipped with a makeshift bed [7]. In June 2008, 31-year-old Anthony Taylor was convicted of child stealing a 15-year-old Tasmanian girl met in WoW. Their relationship grew to involve use of the telephone. Taylor paid for flight and taxi money for the girl to fly to Melbourne and was arrested while waiting for her at the airport [16].

Detective Lt Thomas Kish of the Michigan State Police states, "Child predators are migrating from traditional methods to alternative media. They are going to places where children are." Police are now working undercover in online interactive games in order to try to detect a variety of illegal activities, including grooming of children [12]. Our research attempts to help address this problem by providing a method for automating the detection of hazardous relationship formation between a child and a person met in online games. The concept is to better assist parents in monitoring their children's online activities and making more informed decisions. The resulting software may aid existing cybercrime units [10] and can be used for producing forensic evidence for court if required.

Massively Multiplayer Online games (MMOs) may be used as tools for meeting, socialising and forming relationships with otherwise unknown persons [3]. We define a relationship in a MMO as having involved some communication or some action of dealing with another person. There are three distinguishing elements relative to the analysis of a potentially hazardous MMO relationship with a child:

- the communication potentially exists offline
- the communication is inappropriate
- the communicator has an adult profile.

The framework for automating monitoring and detection of online activities that may lead to offline meetings adds a complementary new approach to child online security, addressing research concepts currently uncharted in the public domain. Inappropriate communications and adult profiling are outside the scope of this research. Methods identified in this research are applicable to automating the detection of criminal relationship formation, grooming, and sharing of offline contact information and/or meeting arrangements.

While some child protection software is currently available, existing software tends to focus on detection of and banning of inappropriate materials that may be viewed, stated or exchanged over the Internet. For example, the Australian Government released the NetAlert Internet content filter [1]. The use of these types of filters has many advantages particularly for younger children, however the list of sites and materials to be banned is enormous and constantly changing. Filtering software such as NetAlert can be bypassed [9] and in any case does not address many of the bigger issues online, for example, the need to identify inappropriate relationships forming and offline meetings being arranged. In addition, the notion of securely restricting teenagers from specific types of Internet use is for some households, impractical.

Habbo [8] is an example whereby filtering of the exchange of phone numbers, addresses, and other personal information occurs; our research intends to monitor and detect rather than filter. CRISP Thinking [5] for ISPs purports to use heuristics and active moderators for text related grooming detection. Their recent products are said to use an engine for content analysis and relationship analysis alerting for social networks and MMOs. The automated alerts are viewed by an external human moderator who may block or ban bullies or predatory communicators. Their

978-0-7695-4138-9/10 $26.00 © 2010 IEEE
DOI 10.1109/ASONAM.2010.66

software is said to detect when the user is revealing personal compromising information. The specific functionality of the system is currently a black-box. Black-box systems have the advantage of unknown methods to those attempting to evade detection, however they also lack peer review of functionality and accuracy. An open approach is less likely to be subjected to possible reverse engineering, design issues, security risks and other issues [26]. While an open approach to design is appropriate, public knowledge of specific rules of a system for criminal activity detection would be counterproductive.

Our research differs from the above as it aims to provide an automated tool without moderators that is adaptable to multiple users and user types. The design does not store user's sensitive data external to their PC. Our method uses heuristics (as proposed in former research [17]) similar to virus detection methodology [19] evolving and changing over time with the environment.

The contributions of this paper are to add to child safety measures in online gaming, to extend previous efforts to automate the detection of online paedophile activities (outlined in [17]), and to provide a basis for forensics where a child has been subjected to paedophile activities. The paper provides a framework to detect offline (real world) meetings being arranged and offline contact information being shared (described in Section II). It offers a prototype to demonstrate the framework (described in Section III), and performs an evaluation (discussed in Section IV).

II. Architecture and Software Design

This section discusses our design concept of a monitoring and analysis tool that detects where a relationship has potential to escalate to an offline environment. Over time and via a variety of online communication modes a communicator may gather sufficient information to physically locate another person. The threat is where a communicator locates a child offline or the child seeks out and locates the communicator. The aim of the design is to detect and alert the parent of potential for these scenarios.

Our framework design involves monitoring communications and storing communications and event data in a database. We then search this data set for indicators of physical locations or meetings being arranged. We call these indicators *tokens*. Tokens include predefined constant strings, and values defined by rules and regular expressions. Individual tokens have preset ranks defined in the database; ranks are relative to either the constant token value or are defined by the rules and regular expressions that generate the token. The analysis and output stage of the framework involves ranking messages with a level of suspicion based on the tokens found in the message and the ranking of messages within the message window (set of messages surrounding the original message within a predefined timeframe). Messages deemed suspicious are alerted to the parent via email or other report output. The output method and analysis required for output are configurable by the parent. A modular and configurable approach to the framework is fundamental to an ever increasing list of MMO games. Fig. 1 illustrates the design framework.

A. Monitoring, Logging and Filtering

The software is physically placed at the end-user position (host-based), the parent inputs each child's Windows account name. The design logs data in real time then parses and analyses the data post-hoc, it does not impact gaming performance. The design utilizes preexisting software tools that output chat logs for games, instant messages, chat room messages, and other communications. Additional useful chat and game event data is extracted by the creation of add-ons for MMO games (for example, our Lua scripts built for additional WoW data extraction, see Section III). Relevant packets are captured, for example, packets related to values dynamically added to the database such as HTTP packets where the URL matches that previously discussed within the game and SMTP packets headers and data whereby the source or destination email address has been previously communicated in-game.

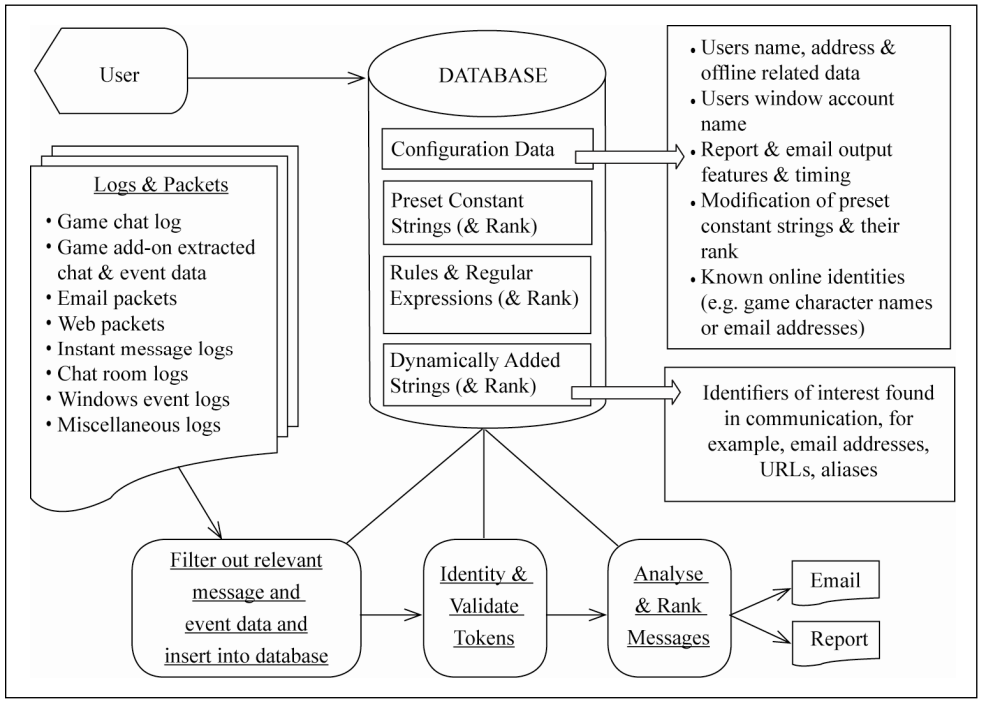

Figure 1: Design Framework

In addition to packets, logs and add-on output data are captured and all data is reformatted and input into the database. Windows event logs are also utilised for determining the user's login and logout times so the system only uses data relevant to the child user(s). Original log data is stored based on date and time, clean up of any old logs and file storage occur at configurable times.

B. Identify and Validate Tokens

A token can be found using a rule, regular expression, or constant string match. Tokens are discarded if they do not comply with validity checks for their type. Some validation methods are configurable dependent on the country, region or language the user defines as their own. Where tokens are found in messages, message event Id, token value, token type, token offset value, and the token's rank are stored in the database. Instead of an ungrouped mass of alerts to the user of suspicious messages, it is possible to group tokens into token types for analysis and output. Tokens can be grouped according to the following types:

- An offline meeting arranged while online. *E.g. string "meet you at" and qualifiers (e.g. date, time).*
- Physical locations of communicators are shared online. *Address formats, suburbs, suffixes, places of interest.*
- Alternative communications to the game are suggested. *E.g. telephone, mobile phone, email, instant messaging, Skype, Ventrilo, TeamSpeak, Chat Rooms.*
- Child User Information. *E.g. the child's name, address, school, clubs, phone and places they frequent.*

Grouping tokens is a configurable output option defined by the user and is a non-fundamental component of the framework. The advantage of grouping tokens into token types is that it provides more legible output and the ability to specify output characteristics. The disadvantage of grouping tokens into types is the crossover of messages deemed suspicious in multiple groups and hence an overlap of output.

Individual tokens are provided a rank as a measure of confidence that the token is a true indicator of offline physical location content, out of game online communications (which indicate other online communications to monitor data for), or an offline meeting being arranged. The ranks are predefined in the database for preset constant strings, rules and regular expression matches. Our method of creating specific ranking values has evolved as a trial and error development process and is similar to spam ranking methodologies in software like the SpamAssassin system [21]. Our design involves positively ranking suspicious tokens and negatively ranking commonly used tokens that are not likely on topic with suspicious communications. For example, in our prototype, the method of ranking involved creating an initial list of commonly used WoW strings from our large sample of "normal game play" data (see Section IV.A. for dataset details) and defines the most commonly used WoW specific strings from this dataset with a negative ranking. An example of positive ranking tokens is whereby our prototype uses regular expression matches to recognise various time and address formats.

The token rank is affected by the degree the token matches a regular expression and multiple possible meanings of matched strings. For example, a phone number matching format with both the prefix (area code) and number values being a valid combination would have a higher rank than a phone number without the prefix. Another example (from WoW chat), the message "this quest is sick... I have to collect tunnel rat ears!" contains the token string "tunnel", a commonly used English word, suffix, suburb, a meeting place and a WoW game related word. The overall rank is slightly positive. The token "quest" is a common WoW word and ranked very negative. The negative outweighs the positive; the overall message is correctly ranked negative.

C. Ranking and Analysis of Messages

During analysis and output, each token type is searched for within the database table and messages with tokens of these types are analysed and ranked; positive ranked messages are then output. Ranking of suspicious messages involves the sum of the token ranks found in each suspicious message along with the sum of surrounding message ranks (within the message window). In summary three levels of ranking exist, level 1 is the individual token rank, level 2 is the message rank, level 3 is the message window rank. The message window time frame can be configured; 2 minutes either side is a practical example.

The speed of analysis is dependent on both the amount of online communications the user has and the amount and types of data the parent chooses to involve in the analysis and output.

D. Alerting/Reporting/Configuration Facilities

The design allows profiling for specific information for the parent to gain an overall picture of the child's online experiences. An example of this is a level of suspicion for each online user (and their various aliases) that the child user communicates with. A list of aliases is updated as the child user meets online identities and aliases are linked together. Other output options include output of analysis of a particular user, of the child's main contacts communicated with, the game play style of the child (for example, listing the main players the child groups with during game play) etc.

The system should be input with the child's/children's personal information related to their physical address, frequented places and other methods of contact (e.g. phone). The parent can enter details of other online identities that are previously known and safe for the child to interact with. In addition, a facility to change ranks for constant strings, rules and regular expression matches is available to assist in the reduction of common false positives.

E. Other Design Concepts and Design Limitations

Similar to a virus protection system, the design incorporates automated updates (for example, changes to rules and token ranking, addition of new modules, etc) via temporary Internet connection to a main server to maintain an up-to-date database and analysis process. The software design requires security of database data and user logs including encryption options, requirements for strong user

passwords, methods to ensure child users do not turn off the software, and various software and data integrity checks.

Limitations of the design include the assumption that detecting linked aliases in MMO games is complete. Existing research demonstrates some methods of detecting aliases between two different pseudonyms (for example [4]) however MMOs have different environmental properties to chat rooms, newsgroups and other online communities due to the high probability that one user is only logged on as one pseudonym at a time. Alternative communications are monitored yet only considered where their presence (such as account names, email addresses, and services) is referenced during the game. The downside of this method is that if a user communicates about for example an email address within the game then sends or receives email related to this address prior to the software being updated this is unmonitored. The alternative method would be to monitor and store vast amounts of unnecessary data packets.

III. PROTOTYPE IMPLEMENTATION

A prototype of the design has been implemented for demonstrative purposes. The prototype is host-based running under Windows XP. The case study game used for the prototype is WoW. The software assumes the parent/guardian of a child/children knowingly installs the software and configures it with user related data (e.g XP account name, home address, language, country, phone numbers, etc). XP login/logout events are obtained using Windows Event Viewer [15] and parsed (with the aid of PsLoglist [18] and prototype scripts) and inserted into the database. Other packets (e.g. email, chat room, IM, and web packets) are logged using Wireshark [23] packet logger.

Lua is a scripting language allowed by the WoW game to run parallel to the game to affect the client interface. While some Lua functionality is restricted by WoW when used in this manner, there remains sufficient functionality for our prototype scripts to gather communication data and properties additional to existing game chat logs and store these game communications properties, chats and events in output files (Lua SaveVariable files) in the prototype database format. Perl scripts were created to extract any required packet data and parse these to database format.

Game information stored by the prototype Lua scripts includes login/logout game events, whisper chat (one-to-one communications), group chats including party chat (up to 5 persons including user), raid chat (communications across multiple party groups grouped as a raid), battleground chat (specific raid chat), and guild chat (guilds are groups of persons with mutual interests formed under a guild name), group join/leave events, guild events including joining and leaving guilds, and guild member login/logoff events (these are used to ascertain guild members online at time of guild messages) and area chat (chats conducted across game based geographical areas including yell, say, tell, emote and all default channel communications such as trade channel, general channel, local channel and other viewable communication channels by the user).

After data gathering, token identification occurs. Our prototype rules and regular expressions include:

- Phone. A sample selection of possible Australian expression formats was created. A validity test is performed on matching values to ensure they also match valid area code and exchange numbers [22].
- Mobile phone (regular expression match created).
- Address street. Match for street number, name and suffix expression format. A validity check is performed using an American list of suffixes [6].
- Time (sample of possible expression formats created).
- Postcode. Valid Australian postcodes [11] are identified if a valid suburb, city or state also exist in the message (postcodes alone create significant false positives).
- Email address (regular expression match created).
- User constants defined by parent user (child offline personal information and account details).

The prototype uses a large table of words to match against whereby each word has an identifier indicating its word type(s). At present the prototype wordlist used includes 17448 words where only A-D are a thorough sample (54% of the wordlist are A-D words). The prototype includes the following subset of possible word types (unless otherwise stated lists were created by researcher):

- General English. From [2, 13] and researcher input.
- WoW. Words from researchers WoW communications was created and manually analysed for most used words with specific WoW meaning. Due to the volume of data, only A-D were used from the list as a sample for the prototype. Other WoW words from [14,20,24,25].
- Suburb. Australian suburbs [11].
- Suffix. American list of suffixes [6].
- Phone (eg “call me”, “my number is” etc)
- Day (eg “Monday”)
- Time (eg “afternoon”, “morning”)
- Place (eg “KFC”, “mall”)
- Other Communication Methods (eg “Skype”, “Xfire”, “email”)
- Meeting (eg “meet at”, “lets meet up”)
- State (Australian states and their abbreviations).

It is possible to deliberately negatively rank a message with a suspicious token in it by also having many negatively ranked tokens in the message. To counteract this the prototype ranks tokens that are highly suspicious (for example, the child’s phone number) extremely high (for example, 1000 where the normal ranking lies between –20 and 20) in comparison to a commonly seen negative ranked token (for example “quest” ranked -15) and within our dataset this significantly reduces this evasion technique.

The prototype output includes lists of suspicious messages sorted into token groups. The prototype output also produces a list of suspicious users and their ranks. The user name (and its aliases) is assessed by a count of suspicious messages, coincidentally suspicious messages, and tokens found within suspicious messages and their surrounding messages. To gather the users overall suspicion rank, each suspicious message is related to a set of one or more users

(for example all party members when a party chat message was sent). Negatively ranked suspicious messages are ignored only positive ranked suspicious messages form the overall user suspicion rank so as to reduce a user's ability to hide suspicious behaviour amongst a set of unsuspicious messages.

A. Prototype Limitations

The prototype considers each message and its surrounding messages to create a rank of suspicion; due to this often several messages within one conversation are flagged with high ranks and hence listed in output multiple times. To resolve this it is possible to rank messages within a distinct time based window as a distinct group uniquely.

When considering a message window, the prototype does not yet differentiate different types of communication and different persons involved in these types. For example, a suspicious communication is stated using *Party* chat, the message window to be analysed should exclude *Guild* chat if the user in the suspicious *Party* chat communication is not a guild member (and hence is not involved in *Guild* chat).

When the child user states offline information to a group, all members should have their suspicion rank heightened however when a non-child user provides offline information to a group chat the child is in, only this communicator's rank should be heightened (i.e. the other party members need not be considered more suspicious). Currently the prototype heightens the ranks of all non-child users involved.

Matching phone, postcode, suburbs, and streets can be further enhanced by gathering more encompassing lists including purchasable lists of phones and addresses from commercial enterprises.

IV. EVALUATION

This section of the paper discusses the data used in the prototype and provides an evaluation of the output from the prototype.

A. Data Sets

The researcher has gathered data from their personal game play experience and online communications (1191 hours over 3.5 years). From this 118,230 chat messages and 922 aliases exist in the prototype database. Some Hotmail, SMTP mail, and Microsoft Messenger Instant Messages were logged. Real data was used to assist in the creation of the framework including strings, rules and regular expressions. This data has been used to highlight any specific issues/intricacies to the concept that otherwise would not have been identifiable using synthetic data. The data does not and is not intended to contain any illicit activities; synthetic data was created for that reason.

Three real cases (presented in Section I) showed persons meet online in the WoW game and activities escalated to arrests. One case implied a child has provided information related to their home address. Another case implied the phone number of either the child or another player was disclosed. Using these cases as examples the aim is to detect when a phone number is provided or home address related details are disclosed. The following synthetic scenarios demonstrate the prototype's ability to detect these types of events:

- Scenario 1: Arranging an offline meeting with user *S1_user* at a physical location near the child's house while online in WoW via online whispers.
- Scenario 2: A user *S2_user* gives the child their phone number while in a WoW grouped situation.

Synthetic communication has been created to demonstrate these two scenarios and has been inserted amongst the researcher's real data. A successful result would be these messages being distinguished as suspicious from amongst real non-suspicious data, as well as the *S1_user* and *S2_user* being distinguished as particularly suspicious users comparatively to non-suspicious users amongst the data.

B. Experimental Results

1) Scenario 1

Table I shows synthetic communication between the child user and *S1_user*. In our example of the prototype's function, the message with Chat ID 95745 is flagged as a key suspicious message by the prototype system as it contains a match with a user constant value (*Margaret*, the child's street). Note that all messages except for that with chat ID 95746 contain one of or a combination of places, days, times, addresses, or meeting arrangement statements and were therefore all output as suspicious messages.

Table II displays HTML for chat ID number 95745 where the user constant matched messages are listed. Table III displays the HTML output related to this communication (i.e. shown when the hyperlink labeled 95745 is clicked in Table II). This message is ranked highly positive.

2) Scenario 2

Table IV shows synthetic communication between the child user and *S2_user*. The message with ChatID 95754 is flagged by the system due to a phone number regular expression match. The phone area code and exchange are individually valid and are also a valid combination. The system finds the players related to the suspicious message and its surrounding communication. When the recipient is *Party* then the system discovers party members at the time of the message. Table V shows one row in the table of HTML output that occurs due to message with ChatID 95754. Table

TABLE I. SCENARIO 1

Chat ID	Instigator	Recipient	Chat Message
95743	Child	S1_user	wanna hook up and go to movies on the weekend?
95744	S1_user	Child	yeah ok, Fri ive got band practise but Sat arvo is good. where will we meet?
95745	Child	S1_user	I can meet you at the end of my street, its Margaret Street. there's a Hungry Jack's there, do you know it?
95746	Child	S1_user	can u get there?
95747	S1_user	Child	yeah should be fine, lets meet at say 2pm and check out what movies are on when we get to cinema

VI displays the HTML output related to this communication (i.e. shown when the hyperlink labeled 95754 is clicked in Table V).

TABLE II. USER CONSTANTS MESSAGES (HTML OUTPUT)

Chat ID	Instigator	Recipient	Message	Total Rank (total amt of tokens found)
95745	Child	S1_user	I can meet you at the end of my street, its Margaret Street. there's a Hungry Jack's there, do you know it?	1341 (16)

TABLE IV: SCENARIO 2

Chat ID	Instigator	Recipient	Chat Message
95748	S2_user	Party	Got Skype?
95749	Child	Party	Nope
95750	S2_user	Party	Xfire?
95751	Child	Party	nah sorry
95752	S2_user	Party	phone?
95753	Child	Party	sure, whats your number?
95754	S2_user	Party	03 9247 6666
95755	Child	Party	kk, I'll ring you in a sec

TABLE V: PHONE/MOBILE MESSAGE (HTML OUTPUT)

Chat ID	Instigator	Recipient	Message	Total Rank (total amt of tokens found)
95754	S2_user	Party	03 9247 6666	2045 (7)

3) Suspicious Users:

Table VII shows the top 7 suspicious user ranks for all real data along with the synthetic data. The two users from the two scenarios were listed at position numbers 2 and 6. *S1_user* shared the same name as other aliases and these aliases had other suspicious messages within the real data. *S2_user* was involved in multiple other suspicious messages most of which were negatively ranked. *Listener* was involved in the one suspicious conversation that in total had 6 positively ranked suspicious messages. Users *Other1, Other2 and Other3* are known suspicious users. *Gamehost* is an instigator of a trivial pursuit type game that occurred twice during the collection of real WoW data. The trivial pursuit game included multiple suspicious tokens for example, cities. The game was played for several hours and the resulting suspicion table reflects the impact of many slightly positive suspicious messages.

C. Discussion

The prototype successfully achieved its objectives: suspicious scenarios were output and highlighted with high ranks, and users from the synthetic data set were shown in the top 7 suspicious users for the entire real data. From the researcher's dataset false positives are summarised in Table VIII. The researcher viewed the messages flagged manually to determine if they were meaningful messages to notify the parent of, meaningless messages are termed false positives. Note that overlapping occurs in the table where messages are flagged in multiple token type groups. By calculating the average sum of the researcher's daily communication records on any given day that some chat communication existed and multiplying by 7 (we assume a child user communicates everyday), an estimated weekly amount of communication is

TABLE III. HTML OUTPUT DESCRIBING MESSAGE WITH CHAT ID 95745

Chat ID	Instigator	Recipient	Message	Token Type	Token Value	Token Offset	Rank	Subtotal Rank
95743	Child	S1_user	wanna hook up and go to movies on the weekend?	128	Movies	24	10	70
				32	weekend	38	10	
				512	hook up	6	50	
95744	S1_user	Child	yeah ok, Fri ive got band practise but Sat arvo is good. where will we meet?	32	Fri	9	10	80
				64	arvo	43	10	
				32	Sat	39	10	
				512	meet	71	50	
95745	Child	S1_user	I can meet you at the end of my street, its Margaret Street. there's a Hungry Jack's there, do you know it?	8	street	31	10	1130
				users_homeaddress_street	Margaret	43	1000	
				128	Hungry Jack's	70	10	
				users_homeaddress_suffix	Street	52	10	
				512	meet you	6	50	
				512	meet	6	50	
95746	Child	S1_user	can u get there?					
95747	S1_user	Child	yeah should be fine, lets meet at say 2pm and check out what movies are on when we get to cinema	time1	2pm	38	1	61
				128	movies	61	10	
				512	meet	26	50	

TABLE VI. HTML OUTPUT DESCRIBING MESSAGE WITH CHAT ID 95754

Chat ID	Instigator	Recipient	Message	Token Type	Token Value	Token Offset	Rank	Subtotal Rank
95748	S2_user	Listener, Child	Got Skype?	256	Skype	4	10	10
95749	Child	S2_user, Listener	Nope					
95750	S2_user	Listener, Child	Xfire?	256	Xfire	0	10	10
95751	Child	S2_user, Listener	nah sorry					
95752	S2_user	Listener, Child	phone?	16	phone	0	10	10
95753	Child	S2_user, Listener	sure, whats your number?	17	number	17	5	5
95754	S2_user	Listener, Child	03 9247 6666	phone2	03 9247 6666	0	1000	
				phone1	9247 6666	3	1000	2000
95755	Child	S2_user, Listener	kk, I'll ring you in a sec	16	ring	9	10	10

TABLE VII. TOP 7 USERS IN SUSPICIOUS USERS TABLE

Alias(es)	Total Amt of Suspicious Messages	Amt of Coincidentally Suspicious Messages	Tokens around suspicious messages counted	Tokens around negative suspicious messages ignored	Sum of Users Negative Ranks in Messages Ignored	User Overall Suspicion Rank
Gamehost	1208	12414	1912	19979	-258162	40936
S1_user, S1_alias1, S1_alias2	680	3157	1651	4556	-47603	28902
Other1	735	4270	1011	8077	-108586	19528
Other2	7	50	133	23	-46	17722
Other3	342	1731	568	2855	-34079	13776
S2_user	70	639	43	1071	-14669	12272
Listener, L_alias1	6	30	42	0	0	12270

made and rounded up to 3000 records per week. Based on percentages we estimate around 12 messages flagged and 8 false positives to be an average amount per week. This is of course indicative only.

Our weekly estimated record amount of 3000 chat messages (as calculated for Table VIII) took approximately 6mins to process the token update component of the system and 25secs to perform the analysis and output HTML reports (using an AMD Phenom 8450 2.10GHz PC).

Many of the false positives stem from discussion related to meeting in the game whereby surrounding information is related to real offline content. For example when users say "qld time" for the time to meet up in the game, the Queensland state matches this and therefore the message is falsely identified as containing offline content of interest. Another example is where users discuss places they have been while not online within the same conversation as arranging an online meeting. This occurs frequently due to players stating other commitments to the game while trying to negotiate mutual game play time. Statements that related to WoW game communications and also appeared to be address formats, occurred often. For example, "10 fresh run", "25 rep run", "3/4 the way", and "3 skill pts". More specific analysis could exclude these occurrences.

No known true positives were overlooked. Once during the dataset it occurred where an offline meeting was arranged with a real world friend of the researcher, no offline details were shared (only date, time and suggestion to "drop around"). This was detected yet not output due to insufficient offline content in the message correctly resulting in a low suspicion rank.

TABLE VIII. PROTOTYPE EVALUATION

Evaluation Groups	Messages Flagged	False Positives
Day/Time	77	57
Address	232	179
Phone/Mobile	14	5
Email	7	0
User constants	71	17
Place/Meeting	66	47
Non-Game Communication Devices	14	0
Total for 118,230 records	481	305
Estimated Weekly Total (based on estimated 3000 records per week)	12	8

The overall result was a list of suspicious messages where a large majority was of valid interest, even though some were ultimately declared false positives. Considering the large data set size, long span of time this was gathered in, and the repetition that occurred with messages from the same conversation and across token groups, the small amount of false positives for the prototype is a very good result. Using the prototype software, the synthesized data set, and the researchers existing data, it appears that if an event occurred whereby a child gave out sufficient information to be physically located or contacted offline by a person met online, or if a child arranged a meeting with a person met online, then the software design will detect this. Also based on these elements it appears the amount of false positives is within an acceptable range. The overall proposed framework has been demonstrated to be plausible.

For future research it would be ideal to apply information extraction and data mining technologies to the research problem. The prototype evaluation would benefit if applied to additional cases of communication about offline meetings. Research could advance the methodology for the ranking system. Finally, development of further automated methods to learn from data would be advantageous.

V. CONCLUSIONS

This paper discussed a possible framework for the detection of potentially hazardous relationships being formed with a child playing a MMO. A prototype to demonstrate the framework was introduced and described. Case scenarios similar to real scenarios were used in conjunction with the prototype and demonstrated the framework as a plausible method for detection of potential for offline meetings being arranged with persons met online.

REFERENCES

[1] Australian Government, *NetAlert - Protecting Australian Families Online*, http://www.netalert.gov.au/ (last accessed Sept 2007)

[2] Beale A., *12dicts word lists*, Release 5, June 2007, http://wordlist.sourceforge.net/12dicts-readme-r5.html (last accessed Feb 2010).

[3] Caplan, S., Williams, D., Yee, N., *Problematic Internet use and psychosocial well-being amoung MMO players,* Computers in Human Behavior, Vol. 25, Issue 6, pp. 1312-1319, 2009.

[4] Chen, H., Goldberg, M. and Magdon-Ismail, M. *Identifying Multi-ID Users in Open Forums*. in Proceedings of the 2nd NSF/NIJ Symposium on Intelligence and Security Informatics (ISI 04), 2004. Tucson, Arizona. Springer.

[5] CRISP Thinking, http://www.crispthinking.com/ (last accessed May 2010).

[6] Erle S. D., *Geo::StreetAddress::US*, CPAN, http://www.usps.com/ncsc/lookups/abbr_suffix.txt (last accessed Feb 2010).

[7] GamePolitics.com, *FBI: Pedophile Met Would-be Victim in World of Warcraft (World of Warcraft)*, Jun 2008, http://www.gamepolitics.com/2008/06/25/fbi-pedophile-met-would-be-victim-world-warcraft (Last accessed January 2009)

[8] Habbo, http://www.habbo.com/ (last accessed May 2010)

[9] Higginbottom N. and Packham B., *Student cracks Government's $84m porn filter*, Aug 2007, http://www.news.com.au/dailytelegraph/story/0,22049,22304224-5005941,00.html (last accessed September 2007).

[10] Hinduja, S., Schafer, J. A., *US cybercrime units on the world wide web,* Policing: An International Journal of Police Strategies and Management, Vol. 32, No. 2., pp. 278-296, 2009.

[11] homehelp4u, *Postcode Tool - List Postcodes, Cities and Suburbs*, http://www.homehelp4u.net/postcode_tool/postcode_list_VIC.php (last accessed May 2010).

[12] Koch W., *Predators use gaming consoles to 'get foot in the door'*, USA Today, 7 Feb 2008, http://www.usatoday.com/tech/news/2008-07-01-porn_N.htm (Last accessed Jan 2009)

[13] Linux, English and British Dictionaries, /usr/share/dict/words (last accessed Feb 2010).

[14] Maw, *World of Warcraft Dictionary*, ForsakenFarmers, http://www.wow-pro.com/wiki/world_warcraft_dictionary (last accessed Feb 2010).

[15] Microsoft, *Understanding Event Viewer*, http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/event_overview_01.mspx?mfr=true (last accessed Feb 2010).

[16] National News, *World of Warcraft pedophile stole teen girl*, June 30, 2008, http://www.news.com.au/story/0,23599,23944622-421,00.html?from=public_rss (Last accessed Jan 2009)

[17] Penna L., Clark A., Mohay G., *Challenges of automating the detection of paedophile activities on the Internet*, First International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE'05), pp 206-22, 2005.

[18] Russinovich M., *PsLoglist V2.7*, http://technet.microsoft.com/en-us/sysinternals/bb897544.aspx (last accessed Feb 2010).

[19] Sanok Jr, D. J., *An analysis of how antivirus methodologies are utilized in protecting computers from malicious code,* Proceedings of the 2nd annual conference on Information security curriculum development, pp 142-144, 2005.

[20] Shier, *World of Warcraft Word Meaning and Acronym list*, Unofficial World of Warcraft Forums, January 2008, http://wow.incgamers.com/forums/showthread.php?t=409628 (last accessed Feb 2010).

[21] Spam Assassin, http://www.spamassassin.com/ (last accessed May 2010).

[22] Wapedia, *Wiki: Telephone numbers in Australia*, http://wapedia.mobi/en/Telephone_numbers_in_Australia (last accessed Feb 2010)

[23] Wireshark, http://www.wireshark.org/ (last accessed Feb 2010).

[24] WoWSlang, *World of Warcraft Slang Dictionary*, http://www.wowslang.com/list.php (last accessed Feb 2010).

[25] Xcrucio*, The WOW dictionary*, World of Warcraft Forum, 16 Sept 2007, http://forums.wow-europe.com/thread.html?topicId=771140076&sid=1 (last accessed Feb 2010).

[26] Young, A., Yung, M., *The Dark Side of "Black-Box" Cryptography, or: Should We Trust Capstone?*, CRYPTO-96 , LNCS 1109, pp. 89 - 103, 1996 .